

- Method Name: createFolder
- Function: create a new folder
- Input: contentType - that need to save in the folder
- Output: basePath-new folder path
- Test Use Case:

| Use Case ID | Test Target | ContentType | Expectation Result | Finial Result | Accept |
|-------------|---|-------------|------------------------|--|--------|
| 1 | get correct path when contentType is none | None | ../downloads/2015/8/6/ | TypeError: cannot concatenate 'str' and 'NoneType' objects | YES |
| 2 | get correct path when contentType is "" | "" | ../downloads/2015/8/6/ | ../downloads/2015/8/6// | YES |
| 3 | if directory was exist and contentType are correct , new folder wouldn't cover the old directory. | 'image' | don't cover | don't cover | YES |
| 4 | it's successful to create new directory at first time when the contentType is correct | 'image' | create successfully | create successfully | YES |

- Method Name: geneFileName
- Function: create file name automatically
- Input:name,suffix
- Output:fileNameValue
- Test Use Case:

| Use Case Id | Test Target | Name | Suffix | Expectation Result | Finial Result | Accept |
|-------------|---|--------|--------|----------------------------|----------------------------|--------|
| 1 | get correct file name when the name is "" | "" | "txt" | 2015-8-6-15-38-38.txt | 2015-8-6-15-38-38-.txt | YES |
| 2 | get correct file name when the name is correct and the suffix is "" | "test" | "" | 2015-8-6-15-38-38 | 2015-8-6-15-38-38. | YES |
| 3 | get correct file name when the name and the suffix are correct. | "test" | "txt" | 2015-8-6-15-38-38-test.txt | 2015-8-6-15-38-38-test.txt | YES |

- Method Name: getSequenceResultImage
- Function: create DNA link picture by DNA sequence
- Input:sequence,picture's width,picture's height,picture's name
- Output:filename
- Test Use Case:

| Use Case Id | Test Target | Sequence | Width | Height | Name | Expectation Result | Finial Result | Accept |
|-------------|--|----------|-------|--------|-------|---|---|--------|
| 1 | get correct result when the sequence is wrong | "1_2+_3" | 100 | 100 | first | method can throw exception | throwing exception and doing nothing because it can't deal "2+" | YES |
| 2 | get correct result when the sequence is null | "" | 100 | 100 | first | method can throw exception | throw exception ,literal for int() with base 10:'. . | YES |
| 3 | get correct result when the sequence only have one parts | "1" | 100 | 100 | first | can't change path in Windows | | |
| 4 | get correct result when the width and height below zero | "1_2_3" | -100 | -100 | first | method will find some wrong about width and height, | beacuse creating image is failed,python crash | YES |
| 5 | get correct result when everything is correct | "1_2_3" | 100 | 100 | first | can't find path in windows | | |

- Method Name: getUserProject
- Function:get all object from one user
- Input:userObj
- Output:result about the user object
- Test Use Case:

| Use Case Id | Test Target | userObj | Expectation Result | Finial Result | Accept |
|-------------|---|------------------|--|--|--------|
| 1 | get correct projects when the user have some projects | user | result{ 'isSuccessful':True, 'projects':[{ 'id':1 'name':'project1' 'creator':Bob", 'function':None, 'track':None },,{ 'id':2 'name':'project2' 'creator':Bob", 'function':None, 'track':None }} } | result{ 'isSuccessful':True, 'projects':[{ 'id':1 'name':'project1' 'creator':Bob", 'function':None, 'track':None },,{ 'id':2 'name':'project2' 'creator':Bob", 'function':None, 'track':None }} } | YES |
| 2 | get correct projects when the user have no project | emptyProjectUser | result{ 'isSuccessful':True, 'projects':[] } | result{ 'isSuccessful':True, 'projects':[] } | YES |

- Method Name: formatProjectList
- Function: To concert project list to dictionary
- Input:projectList

- Output:dictionary result
- Test Use Case:

| Use Case Id | Test Target | ProjectList | Expectation Result | Finial Result | Accept |
|-------------|---|---------------------|--|--|--------|
| 1 | get correct result when the project list is correct | <project1,project2> | result{ 'projects':[{ 'id':1 'name':'project1' 'creator':Bob", 'function':None, 'track':None },{ 'id':2 'name':'project2' 'creator':Bob", 'function':None, 'track':None }] } | result{ 'projects':[{ 'id':1 'name':'project1' 'creator':Bob", 'function':None, 'track':None },{ 'id':2 'name':'project2' 'creator':Bob", 'function':None, 'track':None }] } | YES |
| 2 | get correct result when the project list is core | [] | result[] | result[] | YES |

- Method Name: getChainList
- Function:get some links from a project.
- Input:projectId
- Output:dictation result about some link information
- Test Use Case:

| Use Case Id | Test Target | projectId | Expectation Result | Finial Result | Accept |
|-------------|--|-----------|--|--|--------|
| 1 | get correct result when the id is correct | 1 | result{ 'projects':[{ 'id':1 'name':'chain1' },{ 'id':2 'name':'chain2' }] } | result{ 'projects':[{ 'id':1 'name':'chain1' },{ 'id':2 'name':'chain2' }] } | YES |
| 2 | get correct result when the id is none | None | result[] | result[] | YES |
| 3 | get correct result when the id isn't exist | 5 | result[] | result[] | YES |

- Method Name: getChain
- Function:get link information by id
- Input:chainId
- Output:dictation result about part
- Test Use Case:

| Use Case Id | Test Target | chainId | Expectation Result | Finial Result | Accept |
|-------------|---|---------|--|--|--------|
| 1 | get correct result when the id link is exist. | 1 | ('isSuccessful':True 'part_id':1 'part_name':'part1' 'part_type':'type1' },{ 'part_id':2 'part_name':'part2' 'part_type':'type2' },{ 'part_id':3 'part_name':'part3' 'part_type':'type3' }) " | ('isSuccessful':True 'part_id':1 'part_name':'part1' 'part_type':'type1' },{ 'part_id':2 'part_name':'part2' 'part_type':'type2' },{ 'part_id':3 'part_name':'part3' 'part_type':'type3' }) " | YES |
| 2 | get correct result when the link is null. | 2 | (True,[]) " | (True,[]) " | YES |
| 3 | get correct result when the link isn't exist. | 3 | (False,None) | (False,None) | YES |

- Method Name: createProject
- Function:create project
- Input:request
- Output:result about create successfully or not
- Test Use Case:

| Use Case Id | Test Target | request | Expectation Result | Finial Result | Accept |
|-------------|---------------------------------------|--|---|---|--------|
| 1 | create project when user aren't logon | way:post data: {'track':self.track.id} | {'isSuccessful':False} | {'isSuccessful':False} | YES |
| 2 | create project when the user logs on | way:post data: {'track':self.track.id,'name':'project1'} | {'isSuccessful':True 'project_name':'project1', 'id':1, 'track':'track1', 'creator':'Bob' } | {'isSuccessful':True 'project_name':'project1', 'id':1, 'track':'track1', 'creator':'Bob' } | YES |

- Method Name: createNewDevice
- Function:create new device
- Input:request
- Output:result about new device
- Test Use Case:

| Use Case Id | Test Target | Request | Expectation Result | Finial Result | Accept |
|-------------|--|--|---|---|--------|
| 1 | create device when user doesn't log on | way:post data: { 'name': 'newChain', 'id': 1 } | { 'isSuccessful': False } | { 'isSuccessful': False } | YES |
| 2 | create device when user logs on | way:post data: { 'name': 'newChain', 'id': 1 } | { 'isSuccessful': True 'name': 'newChain' 'id': 1, } | { 'isSuccessful': True 'name': 'newChain' 'id': 1, } | YES |

- Method Name: saveChain
- Function:save the change about link
- Input:request
- Output:result about change
- Test Use Case:

| Use Case Id | Test Target | Request | Expectation Result | Finial Result | Accept |
|-------------|--|--|---------------------------|---------------------------|--------|
| 1 | get correct result when the chain_id isn't exist | way:post data:'chain':'1_2','id':3" | { 'isSuccessful': False } | { 'isSuccessful': False } | YES |
| 2 | link can be changed when the chain_id is exist | way:post data:{ 'chain':'1_2', 'id': 1 } | { 'isSuccessful': True } | { 'isSuccessful': True } | YES |

- Method Name: getTracks
- Function:get all the track in database
- Input:request
- Output:result about query
- Test Use Case:

| Use Case Id | Test Target | Request | Expectation Result | Finial Result | Accept |
|-------------|--|---------|---|---|--------|
| 1 | get correct result when there are no track | | { 'isSuccessful': True 'tracks': [] } | { 'isSuccessful': True 'tracks': [] } | YES |
| 2 | get correct result when there are track | | { 'isSuccessful': True 'tracks': [{ 'id': 1, 'track': 'track1' }, { 'id': 2, 'track': 'track2' }] } | { 'isSuccessful': True 'tracks': [{ 'id': 1, 'track': 'track1' }, { 'id': 2, 'track': 'track2' }] } | YES |

- Method Name: getChainLength
- Function:get the length of the link
- Input:request
- Output:query result
- Test Use Case:

| Use Case Id | Test Target | Request | Expectation Result | Finial Result | Accept |
|-------------|---|--|--------------------|---------------|--------|
| 1 | get the length of the link when the link is null | way:get data:{ 'id': 2 } link: "" | 0 | 1 | YES |
| 2 | get the length of the link when the link is normal | way:get data:{ 'id': 1 } link: "1_2_3_4_5" | 5 | 5 | YES |
| 3 | get the length of the link when the link only have one part | way:get data:{ 'id': 3 } link: "1" | 1 | 1 | YES |
| 4 | get the length of the link when the link isn't complete | way:get data:{ 'id': 4 } link: "1_5" | 2 | 3 | YES |

- Method Name:loadDataSet
- Function:Read files and save information to array.
- Input:File Name
- Output:Array Information
- Test Use Case:

| Use Case Id | Test Target | File Name | File Information | Expectation Result | Finial Result | Accept |
|-------------|---|--------------|---------------------------|--|--|--------|
| 1 | get correct output when the data file is not null | testdata.txt | "1 3 4 2 3 5 1 2 3 5 2 5" | [['1', '3', '4'], ['2', '3', '5'], ['1', '2', '3', '5'], ['2', '5']] | [['1', '3', '4'], ['2', '3', '5'], ['1', '2', '3', '5'], ['2', '5']] | YES |
| 2 | get correct output when the data file is null | nulldata.txt | null | [] | [] | YES |

- Method Name:createC1
- Function:create input data that the length of candidate items sets is one
- Input:Original Data
- Output:array information

- Test Use Case:

| Use Case Id | Test Target | Input Data | Expectation Result | Finial Result |
|-------------|---|---|---|--|
| 1 | get correct output when the data file is not null | [['1','3','4'], ['2','3','5'], ['1','2','3','5'], ['2','5']] | [frozenset(['1']),frozenset(['2']),frozenset(['3']),frozenset(['4']),frozenset(['5'])] | [frozenset(['1']),frozenset(['2']),frozenset(['3']),frozenset(['4']),froze |
| 2 | get correct output when the data file is null | [] | [] | [] |

- Method Name:scanD
- Function:to covert the k rank candidate items sets to k rank frequently items sets
- Input:original data,candidate items sets,support degree
- Output:frequently items sets
- Test Use Case:

| Use Case Id | Test Target | Original Data | Candidate Items Sets | Support Degree | Expectation Result |
|-------------|--|---|---|----------------|--|
| 1 | get correct output when original data isn't null;candiate items sets isn't null;support degree is from 0 to 1, | [['1','3','4'], ['2','3','5'], ['1','2','3','5'], ['2','5']] | [frozenset(['1']),frozenset(['2']),frozenset(['3']),frozenset(['4']),frozenset(['5'])] | 0.02 | [frozenset(['1']),frozenset(['2']),frozenset(['3']),fro |
| 2 | get correct output when original data is null;candiate items sets is null. | [] | [] | 0.02 | [] |
| 3 | get correct output when original data is null;candiate items sets isn't null; | [] | [frozenset(['1']),frozenset(['2']),frozenset(['3']),frozenset(['4']),frozenset(['5'])] | 0.02 | [] |
| 4 | support degree < 0 | [['1','3','4'], ['2','3','5'], ['1','2','3','5'], ['2','5']] | [frozenset(['1']),frozenset(['2']),frozenset(['3']),frozenset(['4']),frozenset(['5'])] | -0.2 | [frozenset(['1']),frozenset(['2']),frozenset(['3']),fro |
| 5 | support degree > 1 | [['1','3','4'], ['2','3','5'], ['1','2','3','5'], ['2','5']] | [frozenset(['1']),frozenset(['2']),frozenset(['3']),frozenset(['4']),frozenset(['5'])] | 1.1 | [] |
| 6 | get correct output when original data isn't null;candiate items sets isn null; | [['1','3','4'], ['2','3','5'], ['1','2','3','5'], ['2','5']] | [] | 0.02 | [] |

| | | | | | |
|--|--|--|--|--|--|
| | | | | | |
|--|--|--|--|--|--|

- Method Name:aprioriGen
- Function:To covert k rank frequently items sets to the k+1 rank candidate items sets
- Input: k rank frequently items sets
- Output: the k+1 rank candidate items sets
- Test Use Case:

| Use Case Id | Test Target | Input | Expectation Result |
|-------------|--|--|---|
| 1 | get correct output when frequently items sets isn't null | [frozenset(['1']),frozenset(['2']),frozenset(['3']),frozenset(['4']),frozenset(['5'])] | [frozenset(['1','2']),frozenset(['1','3']),frozenset(['1','4']),frozenset(['1','5']),frozenset(['2','3']),frozenset(['2','4']),frozenset(['2','5']),frozenset(['3','4']),frozenset(['3','5']),frozenset(['4','5'])] |
| 2 | get correct output when frequently items sets is null | [] | [] |

- Method Name:apriori
- Function:get 1 rank,2 rank,3 rank frequently items sets from original data.
- Input:original data, support degree
- Output: 1 rank,2 rank,3 rank frequently items sets
- Test Use Case:

| Use Case Id | Test Target | Original Data | Support Degree | Expectation Result | Finial Result | Accept |
|-------------|---|---|----------------|---|---|--------|
| 1 | get correct output when original data isn't null;support degree is from 0 to 1, | [['1','3','4'], ['2','3','5'], ['1','2','3','5'], ['2','5']] | 0.5 | [[['1'],['2'],['3'],['5']],[['1','3'],['3','2'],['2','5'], ['3','5']],[['3','2','5']]] | [[['1'],['2'],['3'],['5']],[['1','3'],['3','2'],['2','5'], ['3','5']],[['3','2','5']]] | YES |
| 2 | get correct output when original data is null | [] | 0.5 | [[[]]] | [[[]]] | YES |
| 3 | get correct output when original data isn't null;support degree below 0. | [['1','3','4'], ['2','3','5'], ['1','2','3','5'], ['2','5']] | -0.5 | [[['1'],['2'],['3'],['4'],['5']],[['1','2'],['3','4'],['1','3'], ['3','2'],['2','5'],['1','4'],['1','5'],['3','5']], [['1','3','4'],['1','3','5'],['3','2','5'],['1','2','5'], ['1','3','2']]] | [[['1'],['2'],['3'],['4'],['5']],[['1','2'],['3','4'],['1','3'], ['3','2'],['2','5'],['1','4'],['1','5'],['3','5']], [['1','3','4'],['1','3','5'],['3','2','5'],['1','2','5'], ['1','3','2']]] | YES |
| 4 | get correct output when original data isn't null;support degree is from 0 to 1, | [['1','3','4'], ['2','3','5'], ['1','2','3','5'], ['2','5']] | 1.1 | [[[]]] | [[[]]] | YES |

- Method Name:toList
- Function:To concert the sets to array
- Input:Sets
- Output:Array Information
- Test Use Case:

| Use Case Id | Test Target | Test Data | Expectation Result | Finial Result | Accept |
|-------------|--|--|---|---|--------|
| 1 | frozenset sets convert to array | [frozenset(['1']),frozenset(['2']),frozenset(['3']),frozenset(['4']),frozenset(['5'])] | [[['1']],[['2']], [['3']],[['4']], [['5']]] | [[['1']].[['2']], [['3']].[['4']], [['5']]] | YES |
| 2 | set sets convert to array | [set(['1']),set(['2']),set(['3']),set(['4']),set(['5'])] | [[['1']],[['2']], [['3']],[['4']], [['5']]] | [[['1']].[['2']], [['3']].[['4']], [['5']]] | YES |
| 3 | get correct output if the type of the input is correct | [['1'],['2'],['3'],['4'],['5']] | [[['1']],[['2']], [['3']],[['4']], [['5']]] | [[['1']].[['2']], [['3']].[['4']], [['5']]] | YES |

| | | | | | |
|---|-------------------------------|----|----|----|-----|
| 4 | get null if the input is null | [] | [] | [] | YES |
|---|-------------------------------|----|----|----|-----|

- Method Name:toFrozenset
- Function:to convert the original data to the sets which the type is frozenset
- Input:File Name
- Output:Array Information
- Test Use Case:

| Use Case Id | Test Target | Original Data | Expectation Result | Finial Result | Accept |
|-------------|---|---|--|--|--------|
| 1 | get correct output if the type of the input is array | [[['1']], [['2']], [['3']], [['4']], [['5']]] | [[frozenset(['1']), [frozenset(['2']), [frozenset(['3']), [frozenset(['4']), [frozenset(['5'])]] | [[frozenset(['1']), [frozenset(['2']), [frozenset(['3']), [frozenset(['4']), [frozenset(['5'])]] | YES |
| 2 | get null if the input file is null | [] | [] | [] | YES |
| 3 | get correct output if the type of the input is set | [] | [[frozenset(['1']), [frozenset(['2']), [frozenset(['3']), [frozenset(['4'])]] | [[frozenset(['1']), [frozenset(['2']), [frozenset(['3']), [frozenset(['4'])]] | YES |
| 4 | "get correct output if the type of the input is frozenset " | [frozenset(['1']),frozenset(['2']),frozenset(['3']),frozenset(['4'])] | [[frozenset(['1']), [frozenset(['2']), [frozenset(['3']), [frozenset(['4'])]] | [[frozenset(['1']), [frozenset(['2']), [frozenset(['3']), [frozenset(['4'])]] | YES |

- Method Name:toBeOne
- Function:remove repeated words and keep one word
- Input:sets
- Output:array
- Test Use Case:

| Use Case Id | Test Target | File information | Expectation Result | Finial Result | Accept |
|-------------|---|--|--------------------|-------------------|--------|
| 1 | frozenset sets can remove repeated words | [frozenset(['1','2']),frozenset(['2','3']),frozenset(['1','2','3','4'])] | ['1','2','3','4'] | ['1','2','3','4'] | YES |
| 2 | get null if the type of the input is null | [] | [] | [] | YES |
| 3 | set sets can remove repeated words | [set(['1','2']),set(['2','3']),set(['1','2','3','4'])] | ['1','2','3','4'] | ['1','2','3','4'] | YES |
| 4 | array can remove repeated words | [['1','2'],['2','3'],['1','2','3','4']] | ['1','2','3','4'] | ['1','2','3','4'] | YES |

- Method Name:getResult
- Function:get result that parts recommended
- Input:current link,frequently items sets
- Output:the result that parts recommended
- Test Use Case:

| Use Case Id | Test Target | Link | Frequently Items Sets | Expectation Result | Finial Result | Accept |
|-------------|---|-------|---|--------------------|---------------|--------|
| 1 | get correct recommended result when the current link and frequently items sets aren't null | ['1'] | [[['1'],['2'],['3'],['5']], [['1','3'],['3','2'],['2','5'], ['3','5']], [['3','2','5']]] | ['3'] | ['3'] | YES |
| 2 | get correct recommended result when the current link is null and the frequently items sets aren't null | [] | [[['1'],['2'],['3'],['5']], [['1','3'],['3','2'],['2','5'], ['3','5']], [['3','2','5']]] | [] | [] | YES |
| 3 | get correct recommended result when the current link isn't null and the frequently items sets aren null | ['1'] | [] | [] | [] | YES |
| 4 | get correct recommended result when the current link and frequently items sets aren null | [] | [] | [] | [] | YES |

| |
|--|
| |
|--|

| |
|--|
| |
|--|

isAllowAutoLogin

Test Use Case 1

| | |
|-------------|--|
| Test Target | To confirm that user can enter the personal website automatically if the user log on the system and check the "remember me". |
| Environment | mac os x 10.10. **browser |
| | |

| | |
|--------------------|---|
| Input | User name & password |
| Steps | <div>1. Open the browser</div> <div>2. Input the url</div> <div>3. click the "log in"link</div> <div>4. input the user name:Bowen</div> <div>5. click the "log on"button</div> <div>6. click the "log out"button</div> <div>7. input the same url</div> |
| Expectation Result | Skip the logging on interface and enter the personal website. |

Test Use Case 2

| | |
|--------------------|---|
| Test Target | To confirm that user can enter the personal website automatically if the user log on the system and don't check the "remember me". |
| Environment | mac os x 10.10. **browser |
| Input | User name & password |
| Steps | <div>1. Open the browser</div> <div>2. Input the url</div> <div>3. Click the "log in"link</div> <div>4. Input the user name:Bowen</div> <div>5. Input the password:1</div> <div>6. Click the "log on"button</div> <div>7. Click the "log out"button</div> <div>8. Input the url</div> |
| Expectation Result | Skip the logging on interface and enter the personal website. |

Test Use Case 3

| | |
|--------------------|---|
| Test Target | To confirm that user can't enter the personal website automatically if the user check the "remember me"and don't log on. |
| Environment | mac os x 10.10. **browser |
| Input | User name & password |
| Steps | <div>1. Open the browser</div> <div>2. Input the url</div> <div>3. Check "remember me" button</div> <div>4. Close website</div> <div>5. Input the url</div> |
| Expectation Result | Enter the logging on interface. |

Test Use Case 4

| | |
|--------------------|---|
| Test Target | To confirm that user can't enter the personal website automatically if user don't check the "remember me" and don't log on. |
| Environment | mac os x 10.10. **browser |
| Input | User name & password |
| Steps | <div>1. Open the browser</div> <div>2. Input the url</div> <div>3. Don't check "remember me" button</div> <div>4. Close website</div> <div>5. Input the url</div> |
| Expectation Result | Enter the logging on interface. |

Test Result:

<

| Use case ID | Expectation Result | Finial Result | Accept | |
|-------------|--------------------------------|--------------------------------|--------|--|
| 1 | enter the personal website | enter the personal website | | |
| 2 | enter the logging on interface | enter the logging on interface | YES | |
| 3 | enter the logging on interface | enter the logging on interface | YES | |
| | | | | |

| | | | |
|---|-----------------------------------|-----------------------------------|-----|
| 4 | enter the logging on interface | enter the logging on interface | YES |
|---|-----------------------------------|-----------------------------------|-----|

logonAction:

| Use case ID | User Name | Password | Expectation | Finial Result | Accept |
|-------------|---|--|----------------------------|----------------------------|--------|
| 1 | Null | Null | Tips: logon failure | Tips: logon failure | YES |
| 2 | Null | 1 | Tips: logon failure | Tips: logon failure | YES |
| 3 | Bowen | Null | Tips: logon failure | Tips: logon failure | YES |
| 4 | Bowen | 2 | Tips: logon failure | Tips: logon failure | YES |
| 5 | Bowen | 1 | Enter the personal website | Enter the personal website | YES |
| 6 | Bowen | #@ | Tips: logon failure | Tips: logon failure | YES |
| 7 | Bowen | the length of the password is too long | Tips: logon failure | Tips: logon failure | YES |
| 8 | 錦氣杓 | 1 | Tips: logon failure | Tips: logon failure | YES |
| 9 | the length of the user name is too long | 1 | Tips: logon failure | Tips: logon failure | YES |
| 10 | the length of the user name is too long | the length of the password is long | Tips: logon failure | Tips: logon failure | YES |

registerAction:

| Use case ID | User Name | Password | Repeat Password | Expectation Result | Finial Result | Accept |
|-------------|--|--|-----------------|---|---|--------|
| 1 | Null | Null | Null | Tips: logon failure(Use r can't be null) | Tips: logon failure(Use r can't be null) | YES |
| 2 | Null | 1 | 1 | Tips: logon failure(Use r can't be null) | Tips: logon failure(Use r can't be null) | YES |
| 3 | Bowen | Null | Null | Tips: logon failure(pas sword can't be null) | Tips: logon failure(pas sword can't be null) | YES |
| 4 | Bowen | 2 | 1 | Tips: logon failure(Use r can't match) | Tips: logon failure(Use r can't match) | YES |
| 5 | Bowen | 1 | 1 | Tips: logon successfully | Tips: logon successfully | YES |
| 6 | Bowen | #@ | #@ | Tips: logon failure(illegal characters exist) | Tips: logon failure(illegal characters exist) | YES |
| 7 | Bowen | the length of the password is too long | same password | Tips: logon failure | Tips: logon failure | YES |
| 8 | the length of the username is too long | 1 | 1 | Tips: logon failure | Tips: logon failure | YES |

| | | | | | | |
|----|--|--|---------------|-----------------------|-----------------------|-----|
| 9 | the length of the username is too long | the length of the password is too long | same password | Tips: logon failure | Tips: logon failure | YES |
| 10 | 銕氣杓 | 1 | 1 | Tips: logon failure | Tips: logon failure | YES |
| 11 | Bowen | 1 | 1 | User name has existed | User name has existed | YES |
| 12 | Bowen | 2 | 2 | User name has existed | User name has existed | YES |